Grid User Requirements – 2004: A Perspective From the Trenches

Steven J. Newhouse and Jennifer M. Schopf

Abstract – **Pervasive Grid adoption is predicated on the availability of widely deployed usable software and a user community willing to use it. Currently, widespread adoption of Grids, even within technically sophisticated communities, is limited, and determining and eliminating these barriers to adoption are essential in order for Grids to becoming widely capitalized. Through a series of face-to-face interviews conducted during the summer of 2004, we have identified issues relating to job submission, file transfer, usability, and systems management that must be resolved in order to improve the usability of Grid infrastructures. The background to these issues and some possible solutions are described in this paper.**

Index Terms – **Grid functionality, security, tools, user requirements**

I. Introduction

Over the last ten years, the use of Grids and e-Science has grown and changed. From very early application-specific bespoke solutions, we now have a variety of general-purpose software and solutions for users to work from. However, it is strongly felt in the community that existing software has not evolved sufficiently to meet changing user needs, especially as the community matures, and additional guidance for future plans is needed.

During July and August 2004 we visited a set of applied science and middleware groups in the UK in order to gather basic information on the services and functionality these projects were using. Our motivation was to help guide the development of future activities and priorities within the UK's Open Middleware Infrastructure Institute [1] and the Globus Alliance [2], and to inform the wider Grid community of the barriers to uptake for Grids and e-Science. We held meetings with application developers with some Grid (generally Globus Toolkit 2 or Globus Toolkit 3) or Web services experience, those with software that might be of broader use or interest, and those who had expressed dissatisfaction with current tools, in order to understand their issues in more detail. The twenty-five groups, listed in Appendix A, included widely used applications from biology, chemistry, physics, climatology, and other scientific fields, as well as a smaller set of basic tool builders. In addition, informal discussions took place at several workshops during this time to get a broader scope in certain areas [3,4,5].

Meetings with groups varied from half-hour to half-day time slots and covered a wide variety of topics, concentrating on current use and needs. We considered performing structured interviews based on a standard questionnaire, but differences in backgrounds and knowledge of the interviewees made this approach ineffective, so the interviews were not formally structured as such. Instead, we asked what functionality the groups had tried in the past, what their applications needed today from the current Grid infrastructures, and what functionality the group was considering for near-future plans. Most meetings ended by our inquiring what functionality the group thought was most important and still lacking in today's tools or services.

Over the course of the interviews several basic ideas began to repeat themselves; and while we cannot claim to have interviewed all possible groups, the topics covered by the second half of our discussions only reinforced the initial data, thereby implying that we had established the significant issues for this particular community.

In this article we summarize the results of our conversations with users. While the results are not unexpected, we note that the ranking of needs by the users was quite different from what many tool developers have assumed. We detail our findings in the areas of the continued need for training, security, service functionality as seen by the users, details on tools, and build/infrastructure comments. In Section VIII we highlight the most commonly stated concerns that emerged from these interviews.

II. The Broad User Perspective and Roles

Within the UK, there has been a strong top-down effort to encourage researchers to take advantage of Grid computing and e-Science tools for research. In 2001, a joint program between the UK's research councils and the Department of Trade and Industry (a UK Government department), totalling £98m, was one of the first national programs to strongly encourage e-Science uptake among the applied research community, and largely responsible for some of the early advances in the field.

Manuscript received Sept 29, 2006. This work was supported in part by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under contract W-31-109-Eng-38, NSF NMI Award SCI-0438372, JISC, EPSRC Grant Reference GR/S90843/01, and OMII: Open Middleware infrastructure Institute.

Steven J. Newhouse is with the Open Middleware Infrastructure Institute, University of Southampton, Southampton, UK, SO17 1BJ, s.newhouse@omii.ac.uk .

Jennifer M. Schopf is jointly appointed at Argonne National Laboratory, Argonne, IL 60439 USA and the UK National e-Science Centre, Edinburgh, UK EH8 9, jms@mcs.anl.gov .

However, at that time, many of the promoted tools were not of production quality, so even years later when we did our survey, many felt that Grids and e-Science did not meet their promise. For many of the groups we spoke to, the vision of the Grid—the use of distributed resources across different organizations—had not yet been fully accepted, let alone implemented. For instance, we encountered some groups that didn't understand why they should consider Grid tools over their usual SSH and scp. Almost all the applied science groups felt that the basic concepts of the Grid, and the maturity of software being developed to support these activities, had been oversold, and they were therefore now much more cautious in adopting new software infrastructures and hesitant when new versions of old tools were released.

This first observation highlights the changing nature of the Grid. As the adoption and deployment of Grid infrastructures matures it is inevitable that we see a partitioning of roles across groups of individuals and the emergence of specialisations within the community as well. Whereas several years ago one person would install 3rd party software, adapt it for use in a specific field domain, and then use the software for their own research, we saw a splitting of these roles in most groups. Common role divisions we saw were:

- *Service providers*, e.g. system administrators, virtual organisation managers, and providers of generic services, e.g. NGS.
- *Generic tool developers*, e.g. people working on general tools such as OMII or the Globus Toolkit,
- *Technologists*, or developers of tools and services for domain-specific technology providers, including alterations to generic tools, and
- Domain specific *end-users*, e.g. scientists and researchers.

The splitting of roles amongst multiple people has had a cascading effect on how e-Science projects function, and as we'll describe in later sections, the different backgrounds of these roles show a need for additional training and interfaces so that each role subset has the tools needed. It is therefore essential that as we continue to promote the uptake of this new paradigm that we do so with an understanding of the comfort zones of each community, which may be contradictory, and that the introduction of this paradigm shift must be done with minimal changes to the current mode of operation.

III. Training and Documentation

All the participants — end-users, technologists, generic tool developers, and service providers — expressed a strong need for basic common practices within the Grid. Although Web services, firewalls, build instructions, and security were top areas of concern, there was an expressed need for better documentation of common practices across all these areas.

Many users, especially the technologists tuning generic software for specific application communities, felt that the documentation provided with Grid software was pitched at the wrong level. For example, for Web services, many of the middleware or application developers we spoke with were looking for a hands-on three-hour approach to understanding the basics—not a high level vision and not low-level tuning. The end-users and technologists we spoke with were more interested in API instruction—how to use the common tools and to understand the tradeoffs among these tools. Example applications that demonstrate the use of these APIs were seen as an essential element of any software infrastructure and much preferred to vague statements of capability. Educational material needs to be targeted more precisely at the particular user roles.

In the last two years, part of the need for training at a variety of levels has been met in the UK by OMII and the NeSC Training team [6], which has offered a wide variety of training varying from WSDL and Web service basics, to induction courses for EGEE [7, 8], OMII [1,9], and the NGS [10]. This has included both lower-level developer training and higher-level outreach, especially for non-traditional communities such as e-Social Science [11] and the Arts and Humanities [12].

In general, the documentation provided by the middleware offerings was perceived to be lacking in detail and accuracy. Particular frustration was expressed about builds and packaging. This frustration highlights an important point. Users, especially those not in a traditional system administration role, are not likely to persevere in trying to use an infrastructure that they have not been able to install. If they are not able to install the software and verify that it is working at some level, they are likely to give up and move on. However, details on specifically what was missing in the documentation were difficult or impossible to discover. The real solution may be to aid in better coordination between documentation writing and groups that are going to be using the material.

Since this survey of user requirements, significant effort by most middleware groups has been spent on better documentation and interactions with users in order to tune the documentation. For example, the OMII project provides distinct tutorial, user, and administrative manuals drawn from contributors and edited by dedicated staff to provide consistent style and content. Within the Globus Toolkit, a documentation specialist was hired, whose sole purpose is to cover documentation and make sure the content is provided in an accessible way that is consistent across components, each consisting of separate user, developer, and administrator documents. In addition, documentation for GT software components is a stand-alone open-source open-contribution project [13] and individual components are graded on their documentation completeness [14].

IV. Security

If security isn't easy to use, end users will find a way to not use it [15A]. Indeed, many groups had no security infrastructure associated with the tools or services in use. Common reasons included the configuration of services on local systems or behind firewalls, the lack of an agreed-upon security infrastructure among all sites, the concern about overhead or the effect security would have on performance, and the lack of control over remote service security. Some of these groups that had adopted certificate-based authentication systems were using short or non-existent pass phrases or were sharing certificates between individuals, effectively rendering these systems insecure. Most groups in this class agreed that this situation was not ideal but was acceptable for the time being, and none of them had near-term plans to alter their security infrastructure.

A. Firewalls

Firewalls continued to be a headache for technologists, general developers, and end users, and a cure-all for service providers. There was general agreement that all stakeholders in network security (the firewall administrator, the local system administrators, and the users) need better instruction on the interactions between firewalls and the commonly deployed services and software. Documents such as the Globus Toolkit firewall requirements document [16] and others [17,18, 19A] provide a solid base on which to build a common practice document describing good ways for system administrators and users to interact over firewalls. While Web services are seen as a way to "drill through" firewalls using commonly opened ports, the introduction of protocol-sensitive firewalls may help eliminate this option and force greater communication among all the stakeholders involved in the network.

The trade-offs between opening more ports for performance, and locking them all down is starting to be recognized as well. However, this remains an open area of research, and practitioners don't seem to have agreed upon best practices at this point in time. For instance, in the last two years, intrusion detection systems such as Bro [20, 21], SNORT [22] and TripWire[23] are being used to supplement firewalls.

B. Delegation

Another area of concern was the need for delegation and the lack of a Web service standard. Delegation is needed in a Grid services environment for third-party transfers, many portal interactions, and even some workflow scheduling approaches.

Within the Grid Security Infrastructure (GSI) approach, delegation consists of the transfer of an IETF-compliant proxy certificate from a client to a 3^{rd} party server. The service on the server may then use this certificate to access other services on behalf of the user. This requires a level of trust by which the host providing the files is willing to provide access to the requesting service.

However, the actual functionality desired by many groups is *limited delegation,* that is, the ability for a user to grant to some third party the right to perform certain limited actions on their behalf but not completely impersonate the user. Proxy certificates do offer a way to do limited delegation through a proxyPolicy field that lets one express what the proxy certificate can do, however this is not in common use.

Limited delegation is not currently in use for two reasons:

1. There is no standard language for expressing a limited delegation (e.g. we have no way of saying "this service can access file Foo and then launch a job that doesn't consume more than 100 SUs as me").
2. Usability: in practice if a user gets the limited delegation policy wrong, their process fails, they get frustrated, and simply throw out the limited part.

For example, GT4 GRAM currently supports "limited proxies" that allow users to delegate a proxy that doesn't allow the recipient to spawn jobs. Even this simple capability has caused usability problems and been disabled by a number of sites.

Two potential standards, WS-Trust [24] WS-Policy [25] are under current discussion, although implementations have not yet been widely adopted. Both of these specifications define frameworks that allow for parties to talk about trust and policy. Similar to the policy field in the proxy certificate, they define the envelope for the conversation, but not the actual language. There is still much work to be done in this area, and in the broader area of roles and responsibilities within Virtual Organisations.

C. General Concerns

The different user roles also expressed different high level concerns with security. System administrators remained primarily concerned with break-ins and malicious users. However domain end-users were much more concerned with data integrity - that data would be corrupted or lost. Of course, in projects dealing with medical records, considerable restrictions existed on who may view patient records, including medical images and written notes.

In addition to these general concerns, users identified several tools needed in the security area:

- A tool to verify the network connectivity between clients and services to ensure that any firewall configuration changes (or other network alterations) would not inhibit established user activity.
- A tool to verify that a service was secure, which in most cases meant that the messages being passed were encrypted or that no one could erase needed data from a system.
- Security audit tools to verify that patches and known exploit-prevention software were still functioning properly after upgrades or changes to the system.

V. FUNCTIONALITY

Almost every group we spoke with was using Grid environments to support their applied science activities, so the functionality they wanted was for their day-to-day work, not farther-out speculative needs. These primarily fell into two categories: job submission and tracking, and file transfers. A few projects were using tool 'add-ons', such as visualization tools, data format translators, or policy management tools, but these were always strongly tied to the project domain and narrowly scoped. When asked about other possible functionality or services that could be used, we were told these were not on the six-month horizon most groups were currently considering (see Section V(c) for further details).

A. Job Submission and Tracking

Job submission was the most common first service in use by the projects we spoke with. For most projects, this was a simple, dependable, "run my application" interface that was in the users "comfort zone" and behaved as expected. Different users defined their comfort zones differently; indeed, most middleware developers believe the phrase "behave as expected" to be nondeterministic. However, every group we spoke with was performing at least simple job submission, and many had adapted a standard, 3rd party tool, such as the Globus Toolkit GRAM job submission tool [26] or Condor [27], for project-specific use.

The job submissions were interacting with well-known resources or services. Users were not trying to determine what machine or service to work with at runtime. They had small sets of resources that were used most of the time. Occasionally a user would have policy questions of the resource ("How many free Matlab licenses are there I can use?" "How many jobs can I submit to the queue today"), but the larger discovery questions were not an issue for these users, both for end-user scientists and the project-specific technologists.

The functionality associated with job submission that was most commonly felt to be missing (or in current development by a project) was a way to track jobs once they had been submitted. Most groups reported problems in which a job had been submitted (or a service request had been made) and something had not performed correctly, but they were unable to determine where, why, or how to fix that problem. Groups admitted that they frequently used Grid tools for job submission and file transfers but resorted to SSH and mining log files to debug what was happening on the system. Every group had experienced the phenomenon of a job run completing as expected one day but failing on the next for unknown reasons.

Several tools were identified as needed:

- Tools to aid in failure identification.
- Better logging services to debug failure causes, and debugging paths through the system and for these logs to be centralized (from the user's perspective) to provide a single point to start the investigation.
- Job-tracking services, in general.

In the last two years considerable engineering effort has improved the robustness of the basic functionality being delivered through basic job submission web services. In addition 'simplified' application oriented abstractions to job submission have been defined, such as the Application Hosting Environment and a proliferation of application specific portal environments that hide much of the underlying complexity.In addition, work has begun to better collect log files and to automatically mine them for errors [28]

B. File Transfer

Most users were transferring files using Grid tools such as GridFTP [29, 30] and were happy with the service level they experienced. Some groups needed reliable file transfers, either because they had many small files to transfer and it was easy for one in a thousand to have problems and be left behind, or because they had such large files that the file transfer time was greater than the mean time to failure for some system component, generally the network connection.

A few groups we spoke with were beginning to examine higher-level file transfer services, such as provenance services, access to databases, or replication, but these groups were still primarily prototyping these efforts.

The tool most commonly requested in this space was one to help diagnose the problems, including that of slow performance, seen on systems when performing large file transfers. Performance on WANs had a very high variability because of the many components involved and the variance over time caused by contention. A tool to help users understand where a problem is being caused so they can better understand who to contact was strongly needed.

In the last two years the operational importance of wide-area large scale data transfers, especially with the High Energy Physics community (e.g. EGEE [7], OSG [31]), has driven both the dynamic performance tuning of the underlying data transfers between GridFTP servers and their management interfaces. Additional experimentation around dedicated network connections, e.g. UKLIGHT [32] and Internet2 [33] has demonstrated the potential of dedicated high-performance networks. And some projects, such as the Earth Systems Grid (ESG) [34] have been investigating the use of troubleshooting tools to give errors and warnings on faults, including the MDS4 Trigger Service [35].

C. Other Services

What surprised us most about the tools and services in use by the groups we spoke with was not what they were using but what they *weren't*. Following is a list of services not considered to be essential by a significant majority of this set of users:

- Notification, except perhaps for job progress tracking,
- Registries or resource discovery,
- Reservations, brokering, co-scheduling, or other advanced scheduling techniques,
- Job migration or application checkpointing,
- Accounting and pricing,
- Data migration,
- Instruments.

The reasons for these exclusions are many. Most of the projects we spoke with were hands-on, application-oriented approaches as opposed to research-oriented projects. The software most of them were using was expected to be of production-release quality; they were not interested in prototypes or proof-of-concept software that was not resilient to failures. And in general, the groups were having enough of a challenge getting the basic functionality up and stable, so higher-level services were not considered an immediate priority within the next year.

VI. Tools

In our discussions, all classes of users expressed a wide variety of opinions—often contradictory—about what they would like tools to look like. Most of what was stated came as no surprise. What was surprising, however, was the importance given to some issues.

One of the chief complaints was that many Grid tools offered horizontal functionality and not end-to-end solutions for a given problem. Users, especially the domain specific technologists, requested one-solution tools that would work easily for the 80% rule, and could be used for the other 20% is needed. Of course, what 80% was for one group was not necessarily what it was for another. Also contradictory, we had requests both to have everything needed to use the tool bundled together in one place, and yet to avoid reinstalling software already present on the system.

Many technologists suggested having smaller, simple tools that could be composed together similar to the UNIX piping mechanism. For instance, users wanted to be able to build a workflow by picking and choosing from basic functions, while using small scripts ("shims") to fit between these functions when necessary to translate between mismatched APIs or data formats.

APIs were another topic that users, primarily the end-users, strongly debated. Users agreed that tools should have simple, compact APIs at the user level, and they strongly felt that this API could be different from the one used by a developer if necessary. They also wanted the user API to be in the language or environment of the user's community. For example, if the tool was for biologists, a Perl interface was recommended; if the tool was for the physics community, Python would be more appropriate. In many cases, adding language-specific wrappers to existing APIs would make the tools much more acceptable.

In the last two years standardisation activity with the Open Grid Forum [36] (previously the Global Grid Forum) through the SAGA (Simple API for Grid Applications) Working and Research Groups [37] defined a core set of functionality covering job submission, file management, and other aspects of Grid systems, which is now being mapped to standard language bindings. Simpler but effective are basic wrappers to adapt a complex tool for a user community, with the best example being the tg-cp work in TeraGrid [38] where GridFTP's globus-url-copy has been wrapped to set some performance parameters by default. For example, users can specify –big for large file tuning, and –many for large numbers of files.

VII. System Administration

One of the main problems that service providers and administrators focused on was the lack of reproducible and verifiable builds and the lack of general stability in both builds and tool use in much of today's Grid software. A concern was the large size and complexity of current software distributions that make debugging build failures and any incremental development work on the code base very difficult. With an increasing number of commercial and research organizations offering software components, simplification of build, packaging, and dependency infrastructure was seen as a priority in order to promote interchange.

A. Builds and Upgrades

When building middleware, service providers wanted a hands-off process that would work the same way every time the software was built. In addition, service providers expressed the need for verification tools—ways to be sure that the build was successful and that the full desired functionality was installed properly. They wanted the prerequisites for all modules to be verified at the start of the build process in order to ensure that the build would run to completion. They wanted errors or further configuration actions to be listed at the end of the build output and not reported intermittently during the build process.

Similar needs were expressed for upgrades. Technologists wanted the documentation for an upgrade to include detailed descriptions of the changes to the system and possible incompatibilities, and to be as straightforward as possible, with verification tests similar to those desired for initial builds. Users of all four types recognized that many services were being used "off label," that is, services written for one function were being used in a setting not envisioned by the designer, and hence upgrades were inevitable. What was desired, however, was a more explicit enumeration of the *effects* of the upgrade. Being able to upgrade components within a distribution rather than reinstalling the whole distribution was seen as highly desirable.

Likewise, system administrators wanted the installation of software requiring privileged access to be minimized and separated from the main build process in order to allow different administrative roles to perform different actions, for instance, system configuration, from database configuration and the build user.

A summary of the list of desired tools and improvements includes

- Verifiable builds,
- A way to verify prerequisite software,
- A hands-off process to build,
- Better error handling,
- Clarification on upgrade properties and effects,
- Minimal privileged access requirements.

In response to these demands, which can be summarized by a desire for professional, production-level code support, both OMII and GT made easier builds and installations a priority. The OMII software (which is predominantly Java based) is distributed in compiled binary form and installations are verified on a variety of platforms by using the NMI Build and Test framework developed by the Condor team at UW Madison.

On the Globus side, the installation process now has a well defined set of prerequisites [39] that have been tested for compatibility of versions. The build process can now be done with individual components with the dependencies figured out automatically by the Grid Packaging Toolkit (GPT) [40], although a server-only or client-only distribution is not yet available. General builds consist of two commands (make and make install). The upgrade process has also been simplified. For current versions of the Globus Toolkit, it is guaranteed that interfaces will not change within a major version (eg, 4.0.2 to 4.0.3), and that all eternal interface changes will be fully documented between major versions (4.0.x to 4.2.x, for example).

B. System Stability

In general, one of the strongest requests was for better system stability and understanding of system stability. Far too often an application that runs perfectly well one day would fail the next, frequently for no easily discernible reason. Because many Grid systems involve tens if not hundreds of components, the mean time to failure has decreased significantly, and better monitoring of the background system is needed in order to detect and debug these issues before they affect users. Many systems are running benchmarks of verification suites, but these tests frequently do not “look like” user applications. For example, just because ping is working between two systems does not mean that large file transfers will also function.

End users and technologists expressed a need for tools to help debug why failures happened, and to help determine who to talk to in order to fix them. For example, almost every group we spoke with had had difficulties transferring large files at one time or another, and not known what was going wrong. Simple “common practices” documents or tools to help users walk through the path of their file transfer would go a long way to addressing these issues.

Having established that the system works, users also wished to see how well it works. Those groups not experiencing outright failures still wanted additional information in order to better understand the performance characteristics of their applications.

A summary of the list of tools includes:

- Better verification tools
- Better debugging tools, especially across services
- Walk-thrus for commonly occurring errors
- Performance information for common actions

Within Globus, the only effort in this space has been the GT4 Trigger Service [35], which can do matching on error conditions and send email to system administrators when services fail. This can be used for verification of services being up before runs.

In addition to these efforts, several tools have been built specifically to address the verification of installs and upgrades. The most prevalent of these is likely Inca [41,42], originally developed for the TeraGrid project [38], but now in use by the UK NGS [10], GEON [43], DEISA[44], and CINECA[45]. Inca runs periodic probes at the user-level to test basic software functionality and verify that sites have compatible instillations.

VIII. Overarching Concerns

During our meetings we identified nine open areas of concern that were repeated by many groups. In no particular order, these areas are as follows:

- **Documentation, Training and Education.** In many of the projects we saw computer scientists being used as a source of technical expertise by the applied researchers. Expanded documentation and the provision of national training and education structures is needed to formalise this need beyond the end of any project (Section III).
- **Network Security.** As firewalls become common place, the interaction of Grid middleware with institutional firewalls is seen as extremely challenging, and system administrators, developers, and users all want more information about common practices and current approaches. (Section IV(a))
- **Limited Delegation.** Many current Grid tools need to be able to perform delegation, and the lack of an industry standard or even a well-understood set of tools for Web services is of great concern. (Section IV(b))
- **Job tracking.** Having conquered the initial challenge of job submission using Grid tools, users are now concerned with understanding where a job is in its lifetime, where it is failing, why, and what to do next. (Section V(a))
- **Composability for functionality.** The desire to have tools perform individual functions has been supplemented by the need to be able to compose these functions together in order to achieve a chain of services to solve application specific problems. (Section VI)
- **Wrappers for usability.** Most users want a stable layer (e.g. an Application Programming Interface – API) between them and the software services in order to bring the functionality into their own comfort zone. These wrappers do not add functionality per se but significantly increase the usability and usefulness of a service. (Section VI)
- **Dependable builds.** Software that builds nondeterministically, is hard to install, or doesn’t include verification test suite is seen as unacceptable by today’s users. (Section VII(a))
- **Verification and instability analysis.** With the overall time to failure for Grid components decreasing as their number increases, there is a strong need for better verification and instability analysis to discover and resolve problems before a user happens upon them. (Section VII(b))
- **User-oriented diagnostic tools.** Most diagnostic tools solve problems other than those seen at the user-level. Tools that look like normal user applications and can help an average user diagnose failures are a strong current need. (Sections VII(b) and V(b)).

IX. Related Work and Next Steps

Our roadtrip of 25 UK user groups is not the only attempt to do wide-scale requirements gathering in the Grid space. Two years prior to our work, Fox and Walker [46] produced an exhaustive survey of current uses of Grid technology through discussions with 80 scientists, primarily in the UK but also some European and American groups. This report also includes several useful classification mechanisms.

More formally, Thomas Finholt’s group at Michigan has been working with requirements gathering for both TeraGrid (in progress), NEESGrid[47], and for eScience collaborations in general [48, 49]. Other social scientists are also beginning to examine this space from that point of view [50, 51].

It is not uncommon for specific user groups to perform their own analysis of Grid requirements as well, both through discussion documents and workshops [52< 53, 54, 55].

In our own work, during the last quarter of 2006 we will be looking at what should be done next in the UK for UK e-Scientists by trying to hear and understand the priorities of those already engaged or close to engaged. This will involve additional site visits and an online survey. A summary report is expected to be produced in early 2007, with a workshop to follow at the UK National e-Science Centre.

X. Summary

Over the course of several weeks in July and August 2004 we spoke with 25 UK e-Science project groups about their use of Grid functionality and services. What resulted is a picture of current application and user needs of these services, and some suggestions for ways to move forward. This data is now influencing the directions of both the Globus Alliance and OMII.

Indeed, since August 2004 there has been a focus within the community on improving documentation and the build processes, and the emergence of dedicated training activities within the UK e-Science Programme. Many of the issues raised within the paper are still perceived as 'problems', but generally with much less severity.

The strongest result that came from these discussions was the simple need for on-going conversations between tool developers and users. Grid tool developers must continue to talk and interact with application scientists; without such interaction, the tools are for nothing.

Appendix A: Project Listing

R. Baldock, MRC Human Genetics Unit and NeSC, Edinburgh Mouse Atlas Project
http://genex.hgu.mrc.ac.uk/
The MRC-funded Mouse Atlas provides an international database of mouse embryo data in close collaboration with other mouse resources around the world. The atlas provides a 3D+time spatio-temporal framework for mapping in situ gene-expression data. The data is a combination of 2D and 3D expression patterns and images, which are spatially mapped to allow spatial as well as textual query. Some data requires significant computation for reconstruction, mapping, analysis and visualisation. This project is in the process of moving from a CORBA-based infrastructure to a Web services infrastructure. A primary consideration in the selection of the technology is stability.

M. Baker, Portsmouth, OGSA Testbed
http://dsg.port.ac.uk/projects/ogsa-testbed
The consortium of Manchester, Westminster, Reading, Daresbury Laboratory, Portsmouth, and Southampton is one of two EPSRC-funded projects to deploy and evaluate the Globus Toolkit v3. While many problems were found with this initial deployment, this group aided in resolving them in software and documentation and in development of additional tools for use in UK e-Science projects.

R.Baxter, EPCC, eDIKT
http://www.edikt.org/
The ELDAS component of the SHEFCE-funded eDIKT project uses standard tools, including Eclipse, and infrastructures to provide access to data resources. A partial implementation of the Global Grid Forum's DIAS (Data Integration and Access Standard) Working Group specification has been developed using EJB's hosted within a JBoss container.

N. Chue Hong, EPCC, OGSA-DAI
http://www.ogsadai.org.uk/
The OGSA-DAI project, and its follow-on OGSA-DAIT, provides a reference implementation of the GGF DIAS specification, which defines uniform access to federated data sources (including files) that may be stored in more than one heterogeneous database. Issues facing the project include the ability to delegate actions across different infrastructures and to execute efficient non-file-based transfers between services where the dataflow at the client and the server may be adjusted in response to varying network conditions. OGSA-DAI is funded by the DTI.

D. Chadwick, Salford, PERMIS
http://www.permis.org/
The discussion with Chadwick covered several projects exploring authorization technologies based around PERMIS and SAML, and projects deploying these solutions in order to evaluate their effectiveness. It was found that these technologies (while relatively mature) needed supporting management tools (e.g., a graphical interface to define and manipulate the XML Security Policies) and an attribute authority infrastructure to contain and manage user roles and capabilities, such as such as SIGNET or the Community Authorization Server (CAS). Many of these projects are based around X.509 certificates, but work is also being done to use Shibboleth as a mechanism to undertake inter-organisation authorisation assertions. These projects are funded by JISC and EPSRC.

D. Colling, IC, GridPP2
http://www.gridpp.ac.uk/
GridPP2 is the current focus of Grid activity for high energy physicists in the UK and involves substantial middleware development as well as deployment on a wide variety of testbeds. This software must be able to accept thousands of jobs at a central broker that must also handle file staging from remote replicas. PPARC funds this work.

T. Cooper-Chadwick, Southampton, gYacht/gShip
http://www.soton.ac.uk/~gyacht/
The Southampton e-Science Centre projects G-Yacht and G-Ship focus on delivery of usable performance predicting (such as speed or seakeeping) design tools for yachts and ships through computational modelling on the Grid. Interaction with the Grid resources is through a portal and HiCOG, which provides access to GT2 and Condor-enabled compute resources.

S. Cox, Southampton, GeoDise
http://www.geodise.org/
GeoDise is an EPSRC-funded pilot project developing an infrastructure to support engineering optimization through the evaluation of parameterized designs on the Grid. Its main feature is the integration of a Grid-capable Matlab. This

functionality covers three primary areas, called as toolboxes: compute (creating a proxy, launching a job, etc.), data (enabling files to be archived, queried, and retrieved through file based metadata), and the conversion of Matlab data structures into XML and vice versa. Generic Web services can also be imported into the environment and invoked from within Matlab.

M. Daw, Manchester, Access Grid & MUST
http://www.agsc.ja.net
http://www.sve.man.ac.uk/Research/AtoZ/MUST/
The JISC-funded Access Grid Support Centre (AGSC) will provide support for UK AG deployments and central services such as an IG Recorder and IG Pix (from inSORS), a virtual venue server and an H323 bridge. The MUST (Multicast Streaming Technology) project is exploring reliable multicast protocols to support Grid applications.

W. Emmerich, UCL, eMinerals and OGSI Testbed
http://eminerals.org/
As part of the e-Minerals project, Condor pools at UCL, Cambridge, Bath, and Reading have been linked by using Condor-G and the Globus Toolkit v2. Federation of Condor pools using Web services is now being explored through a Core programme-funded project in collaboration with the Condor team. Resources are orchestrated through a server-side Business Process Execution Language (BPEL) engine that invokes remote services.

M. Ghanen, Imperial, DiscoveryNet
http://www.discovery-on-the.net/
The DiscoveryNet project, funded by EPSRC, is motivated by knowledge discovery within environmental modelling, geohazard modelling, and gene expression array chips. It integrates different distributed data sources (databases, Web sites, etc.) into a single workflow using a graphical paradigm. Once defined, this workflow can then be "published" and made accessible through a portal.

M. Giles, Oxford, gViz
http://www.visualization.leeds.ac.uk/gViz/
The gViz project enables the visualization of a computational fluid dynamics simulation statically and spatially partitioned by using Metis on a clustered resource from a remote location. MPI is used to communicate between clusters. This work also uses gSOAP, a package that exposes C code through a WSDL interface, to provide a message passing layer, similar to PVM, between nodes. DTI and EPSRC fund this work.

S. Lloyd, Oxford, eDiamond
http://www.ediamond.ox.ac.uk/
Funded by DTI, eDiamond uses GT3 and OGSA-DAI to expose medical records for viewing and analysis between different medical centres. The primary challenge within the project is dealing with the security issues relating to the viewing of medical data. The deployment of this infrastructure on remote resources has been simplified by the use of scripts to ensure repeatable hands-off installation and configuration.

C. Goble & N. Sharmen, Manchester, myGrid and Integrative Biology Project
http://www.mygrid.org.uk/
http://www.integrativebiology.ox.ac.uk/
The myGrid project supports a variety of biology experiments on the Grid, with a focus on semantic properties. A strong element of the project is the integration of different processes (e.g., invocation of a BLAST query on a remote server or looking up sequence information from a database) into an analysis workflow by the user. The provenance of this workflow is recorded, allowing any derived results to be fully described and recreated at a later date. myGrid makes use of and has also contributed significantly to the development of the Taverna workbench, the Freefluo workflow enactment engine and the Scufl workflow language. These projects are supported by EPSRC and BBSRC.

J. MacLaren and J. Brooke, Manchester, Brokering activities at Manchester Computing
http://uombroker.sourceforge.net/docs/server/overview-summary.html
Brokering is an important area in Grids for which there are many experimental solutions but few production services. One such broker, developed by the University of Manchester initially under funding from EUROGRID and GRIP, works with the Unicore infrastructure and is now being further developed within European projects such as UniGridS and DEISA (Distributed European Infrastructure for Supercomputing Applications).

A. Martin, Oxford, ClimatePrediction.NET
http://climateprediction.net/
ClimatePredicition.NET, funded by NERC, uses a structure similar to SETI@Home to simulate future possible climates. A desktop client retrieves an initial data set from a central server to initiate the simulation. On completion the final climate model is uploaded to one of several servers. These servers are accessible by the scientists attempting to derive knowledge from the simulations by linking the initial conditions and final solution.

M. McKeown, Manchester, OGSI:Lite and WSRF:Lite
http://www.sve.man.ac.uk/Research/AtoZ/ILCT
The PERL-based containers OGSI:Lite and WSRF:Lite have been used to develop persistent services and to expose applications as services. C-based clients interact with the WSRF:Lite container using gSOAP before accessing a C-based service wrapped in PERL. The container can be deployed on platforms that do not support Java and can be run through CGI under Apache.

Andy McNab, Manchester, GridPP2
http://www.gridpp.ac.uk/
During GridPP1 considerable work was done to develop several elements of access control infrastructure. One activity, GridSite, required the development of GACL (Grid Access Control Language) and supporting libraries. This language has

been reused in several projects with interfaces from Python, Perl, C, and C++.

S. Pickles, Manchester, TeraGyroid and GRENADE
http://www.realitygrid.org/TeraGyroid.html
http://mrccs.man.ac.uk/research/grenade/
The GRENADE project has used a plugin mechanism within the KDE desktop to provide access to remote compute resources on the Grid for GT 2.4. The scientist is able to use the familiar "drag and drop" paradigm to launch jobs on remote resources and integrate remote files spaces into the local disk space.

A. Porter, Manchester, RealityGrid, and M. Rider, Manchester, eViz
http://www.eviz.org/
The RealityGrid and eViz projects, both funded by EPSRC, are concerned with the running and steering of physical simulations, and providing on-line visualization of application data. Pre-Web service components of the Globus Toolkit are used to support third party data transfers into and out of the computational resource. Job execution, monitoring, and debugging are performed by using GRAM (although the use of other systems such as Unicore have been explored) and SSH to access standard output and error logs.

A. Rector, Manchester, CLEF
http://www.clinical-escience.org/
The MRC pilot project CLEF is investigating how clinical care can be linked to post-genomic databases to add gene based reasoning into the treatment process. A key challenge is to ensure that the clinical records have to be cleansed of patient information before being exposed to the other services. The resulting infrastructure will build on elements of MyGrid (e.g., workflow, portal and Web services).

R. Sinnott, Glasgow, BRIDGES
http://www.brc.dcs.gla.ac.uk/projects/bridges/public/people.htm
BRIDGES (Bio Medical Informatics Grid Enabled Services) is a two-year DTI-funded project being used to explore authorization to medical data. This is one of several security-related projects including DyVOSE, which is exploring the dynamic delegating of trust when issuing certificates. More recent work has provided a GT3 interface to running BLAST on ScotGrid resources.

L. Smith, EPCC, QCDGrid
http://www.epcc.ed.ac.uk/computing/research_activities/grid/qcdgrid/
The PPARC-funded QCDGrid provides a distributed data store with some interactive analysis capability at four sites for the UK QCD community. GT2.4 software including replica management tools was extended to enforce project-specific policies. Work is now beginning on defining common interfaces to enable interaction with other QCD-based Grids in the United Stated.

T. Sloan, EPCC, INWA
http://www.epcc.ed.ac.uk/inwa/
The Innovation Node: Western Australia (INWA) project was funded by the ESRC Pilot Projects in e-Social Science programme to inform businesses and regional policy regarding Grid computing. This project involves the integration of private commercial data with publicly available datasets using a Grid infrastructure between the UK and Australia consisting of Globus Toolkit 2 and 3, Sun Grid Engine, Transfer-queue Over Globus (TOG), OGSA-DAI, and the Grid data service browser from the FirstDIG project.

L. Yang, B. Yang, NeSC, AI Workflow
http://dream.dai.ed.ac.uk/e-Science/
AI Workflow, which is in the early stages of research, aims to use proof planning technologies to define and map workflows to resources. The longer-term goal is to use this work to define and exploit different qualities of service to different resources.

Acknowledgment

We would like to thank all the people we visited for their hospitality and the uninhibited technical discussions. We are also grateful to Von Welch for extended conversations over the nature of delegation.

References

[1] Open Middleware Infrastructure Institute," http://www.omii.ac.uk
[2] The Globus Alliance, http://www.globus.org .
[3] "UK Workshop on Grid Security Practice", Oxford, July 8–9, 2004, http://e-science.ox.ac.uk/events/security-workshop/ .
[4] "Service Grids: Current Activity & Middleware Requirements," NeSC Edinburgh, UK, July 22–23, 2004, http://www.nesc.ac.uk/action/esi/contribution.cfm?Title=415.
[5] "Networks for Non-Networkers Workshop," University College London, UK, July 13–14, 2004, http://grid.ucl.ac.uk/NFNN.html .
[6] Edinburgh e-Science Training, Outreach, and Education, http://www.nesc.ac.uk/training/ .
[7] EGEE: Enabling Grids for e-Science, http://www.eu-egee.org/ .
[8] Training Events for EGEE, http://www.egee.nesc.ac.uk/schedreg/ .
[9] UK Training Events, http://www.nesc.ac.uk/training/events/index.html .
[10] UK National Grid Service, http://www.grid-support.ac.uk/ .
[11] NCeSS: The National Centre for eSocial Science, http://www.ncess.ac.uk/
[12] AHESSC: The Arts and Humanities e-Science Support Centre, http://www.ahessc.ac.uk/ .
[13] GT Release Manuals Project, http://dev.globus.org/wiki/GT_Release_Manuals
[14] GT Manuals Quality Grading, http://dev.globus.org/wiki/GT_Manuals_Quality_Grading
[15] Mike Surridge, "Rough Guide to Grid Security", 2002, http:// www.nesc.ac.uk/esi/events/121/RoughGuidetoGridSecurityV1_1.pdf
[16] V. Welch, "Globus Toolkit Firewall Requirements, version 5," July 2003. http://www.globus.org/security/firewalls/Globus%20Firewall%20Requirements-5.pdf
[17] Jon Hillier, "The Use of Firewalls in the UK e-Science Grid: ETF Level 2 and beyond", 2002, http://e-science.ox.ac.uk/events/firewall-workshop/FirewallIdeas.pdf.
[18] Paul Jeffreys and David Boyd, "UK Grid Firewall Workshop Report", , November 2002, http://www.nesc.ac.uk/talks/121/GridCSfirewallworkshopnov5.pdf
[19] Arun Chhatpar, "Managing a grid, Part 2: Security considerations" IBM DeveloperWorks, 24 Oct 2006, http://www-128.ibm.com/developerworks/grid/library/gr-manage2/
[20] Bro Intrusion Detection System, http://bro-ids.org/
[21] V. Paxson, "Bro: A System for Detecting Network Intruders in Real-Time", Computer Networks, 31(23-24), pp. 2435-2463, 14 Dec. 1999.

[22] Snort: The De Facto Standard for Intrusion Detection/Prevention, http://snort.org/
[23] TripWire Intrusion Detection Software, http://www.utexas.edu/its/sds/products/tripwire.html
[24] Anthony Nadalin, Marc Goodner, Martin Gudgin, Abbie Barbir, Hans Granqvist, eds. "WS-Trust 1.3", OASIS Committee Draft 01, #ws-trust-1.3-spec-cd-01 September 6, 2006. Available from http://docs.oasis-open.org/ws-sx/ws-trust/200512
[25] Anthony Nadalin, Chris Kaler, Ronald Monzillo, Phillip Hallam-Baker, eds., "Web Services Security: SOAP Message Security 1.1 (WS-Security 2004)", OASIS Standard Specification, 1 #wss-v1.1-spec-os-SOAPMessageSecurity, February 2006. Available from http://docs.oasis-open.org/wss/v1.1/
[26] Grid Resource Allocation and Management (GRAM), http://www.globus.org/gram
[27] The Condor Project, http://www.cs.wisc.edu/condor/
[28] David Cieslak, Douglas Thain, Nitesh Chawla, "Troubleshooting Distributed Systems via Data Mining", Proceedings of the IEEE Symposium on High Performance Distributed Computing (HPDC), June 2006.
[29] GridFTP, http://gridftp.globus.org.
[30] W. Allcock, J. Bresnahan, R. Kettimuthu, M. Link, C. Dumitrescu, I. Raicu, I. Foster. "The Globus Striped GridFTP Framework and Server", Proceedings of Super Computing 2005 (SC05), November 2005.
[31] Open Science Grid, http://www.opensciencegrid.org/
[32] UKLight: The UK Facility for Supporting Optical Networking, http://www.uklight.ac.uk/
[33] Internet2, http://www.internet2.edu/
[34] ESG: The Eart Systems Grid, http://www.earthsystemgrid.org/
[35] Ann Chervenak, Jennifer M. Schopf, Laura Pearlman, Mei-Hui Su, Shishir Bharathi, Luca Cinquini, Mike D'Arcy, Neill Miller, David Bernholdt, Monitoring the Earth System Grid with MDS4, Proceedings of the Second International conference on eScience and Grid Computing (eScience 2006), December 2006.
[36] Open Grid Forum, http://www.ogf.org
[37] Simple API for Grid Applications Research Group (SAGA-RG), https://forge.gridforum.org/projects/saga-rg/
[38] TeraGrid, http://www.teragrid.org/
[39] GT4 Admin Guide, Chapter 3, Software Prerequisites, http://www.globus.org/toolkit/docs/4.0/admin/docbook/ch03.html
[40] Grid Packaging Tools, http://www.gridpackagingtools.org/
[41] The Inca Test Harness and Reporting Framework, http://inca.sdsc.edu/
[42] Shava Smallen, Catherine Olschanowsky, Kate Ericson, Pete Beckman, and Jennifer Schopf , "The Inca Test Harness and Reporting Framework", Proceedings of SuperComputing '04, November 2004. Also available as SDSC Technical Report #SDSC-TR-2004-3, http://www.sdsc.edu/TR/SDSC-TR-2004-3-IncaTest.pdf
[43] GEON: The GeoSciences Network, http://www.geongrid.org/
[44] DEISA: Distributed European Infrastructure for Supercomputing Applications, http://www.deisa.org/
[45] CINECA, http://www.cineca.it/en/index.htm
[46] Geoffrey Fox and David Walker, "e-Science Gap Analysis", UK Technical Report Series # UKeS-2003-01, June 2003.
[47] B. Spencer Jr., T.A. Finholt, I. Foster, C. Kesselman, C. Beldica, J. Futrelle, S. Gullapalli, P. Hubbard, L. Liming, D. Marcusiu, L. Pearlman, C. Severance, G. Yang, NEESgrid: A Distributed Collaboratory for Advanced Earthquake Engineering Experiment and Simulation, Proceedings of the13th World Conference on Earthquake Engineering, August 2004.
[48] Chuck Huff and Thomas Finholt,, editors, Social Issues in Computing: Putting Computing In Its Place, McGraw-Hill, Inc., 1994.
[49] Spencer, B., Butler, R., Ricker, K., Marcusiu, D., Finholt, T., Foster, I., Kesselman, C., "Cyberenvironment Project Management: Lessons Learned", September 2006, Available from http://neesgrid.ncsa.uiuc.edu/documents/CPMLL.pdf
[50] "History and Theory of Infrastructure: Lessons for New Scientific Cyberinfrastructures Workshop", September 28-October 1, Ann Arbor, http://www.si.umich.edu/InfrastructureWorkshop/
[51] "Information Infrastructures and Architectures Workshop" September 27-28, 2006, , UK National eScience Centre (NeSC), http://www.nesc.ac.uk/action/esi/contribution.cfm?Title=700
[52] R. Baker, "ATLAS Grid Requirements", December 2000, http://www.usatlas.bnl.gov/computing/grid/meetings/ATLAS-Grid-Requirements.doc
[53] "Enterprise Grid Requirements Workshop", GGF18, GridWorld/GGF18, September 11-14, 2006, Washington DC, http://www.ggf.org/gf/event_schedule/index.php?id=444
[54] Oscar Corcho, "Semantic Grid requirements for ontology technology", http://www.cs.man.ac.uk/~ocorcho/InvitedTalks/SMI_July2006.html
[55] Oliveira IC, Oliveira JL, Sanchez JP, Lopez-Alonso V, Martin-Sanchez F, Maojo V, Sousa Pereira A., "Grid requirements for the integration of biomedical information resources for health applications", Methods of Information in Medicine, 2005;44(2):161-7.